Vietnam J Comput Sci (2015) 2:47–56
DOI 10.1007/s40595-014-0030-9

REGULAR PAPER

Obtaining a data quality index with respect to case bases

Jürgen Hönigl · Josef Küng

Received: 14 June 2014 / Accepted: 24 August 2014 / Published online: 11 September 2014
© The Author(s) 2014. This article is published with open access at Springerlink.com

Abstract Within case-based reasoning (CBR), terms concerning quality of a case base are mentioned in publications, but partially without clarifications of criteria. When developing a CBR system from scratch, an index for case base quality supports an assessment of the actual cases. In this approach, both theory and an application are demonstrated. An index was defined and subsequently applied within a proof of concept. In addition, various approaches concerning case base quality are demonstrated. Big data occur within a combination of high velocity, great volume and variety of incoming data. New cases are suitable if they are referring to an economic value. Defining an index to measure the case base quality copes with that. In this paper, an overview of the CBR-related index towards the big picture regarding data quality can be seen. To demonstrate weighting, concrete invocations of the defined subindices are itemized with respect to applied data sets. This paper depicts a generic easy-to-use index with respect to case bases.

Keywords Big data · Case-based reasoning · Data quality · Expert systems

1 Introduction

Within this section, the introduction was divided into several parts to demonstrate the motivation, a few statements about CBR and an outline.

J. Hönigl (✉) · J. Küng
Institute for Application-Oriented Knowledge Processing,
Johannes Kepler University, Linz, Austria
e-mail: juergen.hoenigl@jku.at

J. Küng
e-mail: josef.kueng@jku.at

Noteworthy, this paper extends with various aspects a preliminary version, which appeared as [12]. These extensions can be seen in the related work section when considering the data quality criteria given by [21], an attached section (number 7) that compares other criteria (presented in [21]) with criteria of this CBR data quality index and an additional section (Sect. 8), which presents the possibilities when invoking subindices. Partially, the conclusion section is enhanced with respect to attached new paper content.

1.1 Motivation

When we do literature review about case-based reasoning, it was written about the quality of a case base and avoiding too redundant cases within case base. Various approaches are existing, but partially with fuzzy definitions and primarily without clear results, which are itemized in the related work section. Especially when researching towards an eventual reuse of an index. Therefore, the authors were defining an index to describe case base quality. This was applied within the first author's doctoral thesis as part within the proof of concept, which appeared as [10]. Closing the gap between big data and CBR can be seen as a drive towards an easy-to-apply index for new relevant cases with respect to the size of a case base. The significance of a data quality index can be seen within the next annotations.

1.2 Significance towards a case base

A case base contains knowledge, which will be used for the reasoning process of a case-based reasoning system. An index, which states the quality of a case base, can be used within different steps of the CBR model given by Aamodt

and Plaza [1]. A deletion strategy for too similar cases has to applied to a CBR system to keep the quality of a case base. A deletion strategy is one possible point to deal with the size of case base concerning the maintenance. Another point of view establishes rules for pre-processing to avoid unsuitable reasoning efforts and impaired cases. For instance, a typo could cause an impaired case when not using pre-processing assertion rules. A customer with an age of 92 years (instead of 29 years) could be reasoned within a CBR system, but it would be an outlier within the case base. Subsequently, this case would be removed according to a deletion strategy, which uses the not recently used paradigm for instance. Within CBR, applying an index can be combined with committing a database state. Big data occur if a great volume, a high velocity and variety (structured and unstructured data) will be received. Even two of them can decrease the quality of a case base. A great volume of data with a high velocity can contain too many redundant and obsolete cases. Within a CBR system, pre-processing and similarity measures can avoid many inadequate data, but an assessment of the case base has to be applied in addition. When working on case mining, a complete case base without missing values should be seen as a pre-condition. For instance, gaining association models requires complete cases [13]. When considering an evolution such as IBM's (Industrial Business Machines) research projects Watson and Deep Blue within a decade, it is obvious that these projects can cope with missing values within their knowledge bases [7,19]. In contrast, a CBR approach requires data within the case base because a CBR system is not intended to implement various application programming interfaces to download information on the fly [16]. In addition, the knowledge base of IBM's Watson contained a huge amount of text volumes, databases and journals [8,11].

1.3 Outline

To briefly present a red line regarding this paper, firstly, related work is demonstrated. Then, three subindices are demonstrated, which are required, to build the main index of this approach. Subsequently, the index will be calculated on a top level. Afterwards the application of the index will be explained within a case-based reasoning prototype. Subsequently, a discussion is presented regarding various sights when using thresholds for instance. A comparison depicts the relations of the index presented in this paper towards big picture concerning data quality. Subsequently, an invocation of subindices—that includes weighting—states results for both data sets. In Sect. 9, big data quality explanations are written. At the end, a conclusion and eventual future work are enumerated.

2 Related work

This section demonstrates chronological various possibilities concerning the term case base quality within literature. In 1997, an approach was stated to combine decision theory and CBR. This idea could be used if many missing values would occur to use CBR together with decision theory within an area like unfinished alternatives. Therefore, considering of quality weakness within a case base could be compensated. On the other side, their approach was an experiment and explained difficulties when combing two kind of decision support technologies. For instance, they have detected obstacles when using normative models due to the application of probability and utility for preference and judgement in combination with CBR [25].

A historic approach given in 1998 refers to non-functional requirements regarding CBR systems. Their approach was applied within the medical domain. The efforts made were primarily focused on a CBR system instead of the managed data. An intersection between their system-related approach and a data-related approach can be seen within their work on confidentiality and integrity of data [14].

The quality improvement paradigm (QIP) refers to steps to consider when developing a CBR system. Basili presents a cycle to gain a good combination of technical and managerial solution to achieve a professional CBR application development. The experience factory refers within various steps to different issues, which seems like a waterfall structure at first sight. However, these steps can be partially used in an iterative way, which avoids that. To give a brief explanation concerning this paradigm, two quality-related steps are stated. Within *characterize* (QIP1), the scope of the project will be defined, which results into a context for a goal definition. In addition, experience from the experience base can be selected. The experience base is a knowledge base of past projects related to achieved experience. *Set goals* (QIP2) consider different viewpoints such as customer, project manager and user. The defined goals must be measurable [4].

Within an old approach presented in 2000, quality measures were defined to assess the case base quality with criteria such as correctness, consistency, uniqueness, minimality, and incoherence. They implemented their approach within a framework, but there is a lack concerning eventual other projects when considering application of their approach. In addition, they clearly stated that similarity measures would improve the performance of their assessment. On the other hand, clustering was defined as an issue to perform if their assessment would not be able to process too many cases *in a reasonable amount of time* [23].

Within an approach concerning the maintenance, existing CBR approaches were applied to summarize them into a new approach. On the basis of the Aamodt and Plaza approach, which appeared as [1], and various INRECA research activi-

ties, which appeared as [5], terms were reused and combined. They divided their theoretic generic approach into three stages named retain, review and restore. For instance, retain refers to complete a case. Review points to an assessment of a case and restore implies modifying a case [24] (Table 1).

In [21], criteria regarding the quality of data are stated. Those are demonstrating various aspects, which are briefly itemized to gain the big picture in the data quality domain. In [21], criteria regarding data quality are demonstrated such as—*accessibility, appropriate amount of data, believability, completeness, concise representation, consistent representation, ease of manipulation, free-of-error, interpretability, objectivity, relevancy, reputation, security, timeliness, understandability and value-added*. Within an overview according to [21], they are explained.

Within induction and reasoning from cases (INRECA), case base quality was mentioned, but not concrete stated within a definition of eventual solutions. For instance, a term like *define clear objectives* sounds too unclear to consider it within a concrete index towards case base quality from the author's point of view [5].

Another approach tried to solve and improve maintenance issues with CBR classifiers. They used clustering and logistic regression to build their classifiers. Their approach was not applied within a generic way. Apart of that, the adaptation feature was neglected. Assigning a string label was their *simple adaptation*. When having the focus on maintenance, then adaptation must be carefully integrated into a CBR system from the author's point of view [2].

An approach namely *Assessing Case Base Quality* states interesting notes, but some critical points towards their approach could be seen such as a missing portability and too much effort to integrate their approach. Their main goals were to assess and measure inherent problem-solution irregularity within a case base to improve using cases especially with respect to the accuracy concerning solutions. The Mantel Test (or Mantel's Randomisation Test) was applied together with different ratios to assess the quality of their case base. Therefore, their approach was not implemented in a generic way [22].

Within [20], they stated an approach towards a case-mining algorithm. This generates a *competent* case base when processing existing cases. They stated two issues within their approach. On the one hand, processing nearest cases, which are not containing correct solutions. Another point of view, an uneven case distribution was named as potential obstacle. In addition, they proposed an algorithm to mine within cases, which includes avoiding the previous mentioned problems. Concerning their case-mining approach, they stated two points, which are worth to mention. With respect to the approach in this paper and their approach, their points are referring to issues, which can be mapped to the subindices of this paper [20].

– Each case should cover as much of the problem space as possible to reduce the potential bias, and
– The cases should be as diverse as possible to reduce covariance in producing errors.

When reading these items, a brief comparison to the quality index can be made. The first item above can be seen as avoiding missing values within this approach (third subindex) concerning an index. The second item above can be seen within similar retained queries in this approach (second subindex). In addition, the second item above can be partially seen within the first subindex when assessing average solutions per case.

3 Building subindices

Three indices are used to build an index for the quality of case base. Each of these subindices uses an interval from 0 to 1.

3.1 Index I: average solutions per case

When using a revision graph for solutions, then an entire revision graph will be defined as one solution concerning this index. Null adaptation implies only one solution for a problem, but using a revision graph implies more than one solution for a query. At the end, only one solution is defined as an actual solution for a problem when using a revision graph. Therefore, using revision graphs must not aggravate this index. Multiple solutions are considered as an additional processing effort. In addition, maintenance of a case base can be more difficult with increasing similar solutions. A threshold concerning the maximum number of solutions per case has to be defined within a theoretical interval [1,count of solutions]. A practical interval would be from 3 to 9 due to a ratio concerning solution agility and labor effort [5,18]. For each case, the count of bad solved cases (argument cc), concerning too many solutions, will be incremented if the given threshold was reached. Subsequently, the subindices can be calculated with respect to all cases (argument c).

$$Idx_I = 1 - \frac{\mathrm{cc}}{\sum \mathrm{c}} \tag{1}$$

3.2 Index II: count of similar retained queries

To define similar retained queries, a similarity measure has to be applied with a certain threshold. A problem to problem similarity measure must exist with a known interval to define a threshold for a case base. If a threshold was reached, then the count has to be incremented. Subsequently, an index can be calculated with following formulae:

Table 1 Overview—criterion ⇝ meaning

Criterion	Meaning
Accessibility	The extent to which data is Available, or easily and quickly retrievable
appropriate amount of data	The extent to which the volume of data is appropriate for the task at hand
Believability	The extent to which data is regarded as true and credible
Completeness	The extent to which data is not missing and is of sufficient breadth and depth for the task at hand
Concise representation	The extent to which data is compactly represented
Consistent representation	The extent to which data is presented in the same format
Ease of manipulation	The extent to which data is easy to manipulate and apply to different tasks
Free-of-error	The extent to which data is correct and reliable
Interpretability	The extent to which data is in appropriate languages, symbols and units, and the definitions are clear
Objectivity	The extent to which data is unbiased, unprejudiced, and impartial
Relevancy	The extent to which data is applicable and helpful for the task at hand
Reputation	The extent to which data is highly regarded in terms of its source or content
Security	The extent to which access to data is restricted appropriately to maintain its security
Timeliness	The extent to which the data is sufficiently up-to-date for the task at hand
Understandability	The extent to which data is easily comprehended
Value-added	The extent to which data is beneficial and provides advantages from its use

$$Idx_{II} = 1 - \frac{\text{csrq}}{\sum \text{qc}} \tag{2}$$

The count of similar retained queries is given by argument csrq and the query comparisons are denoted as qc.

3.3 Index III: missing values

The count of missing values (cmv) within cases, with respect to the count of occurrence, has to be calculated.

The actual sum of fields (f) can be achieved within the persistence of a case base when counting all table fields.

$$Idx_{III} = 1 - \frac{\text{cmv}}{\sum f} \tag{3}$$

4 Calculating the main index

To clearly state the formulae, this section presents the integration of the three subindices stated above.

The case base quality index (CBQ) uses an interval from 0 to 100. 100 % states the best possible value for a case base and 0 % refers to a impaired value of a case base. The previous mentioned indices are subsequently weighted.

$$\text{CBQ} = 100 \cdot \frac{Idx_I \cdot \text{Weight}_I + Idx_{II} \cdot \text{Weight}_{II} + Idx_{III} \cdot \text{Weight}_{III}}{\sum_{i=1}^{i=3} \text{Weight}_i} \tag{4}$$

The weight factors can be applied concerning a concrete case base within a given domain. For instance, if avoiding of missing values is more important than the case redundancies, then weight_{III} will receive another argument in comparison to $\frac{1}{3}$.

5 Application of the index within loaner

This section covers the practical aspects of the implementation regarding the index described above. Within code name Loaner, an application written in C# and language integrated querying (LINQ), the approach of this paper was implemented. The visualization was made when using Windows presentation foundation (WPF). The training set of the data was analyzed due to the actual implementation state [9]. It is complete and without multiple solutions, which refers to a good value concerning the case base quality.

5.1 I - Solutions per case

The used threshold for solutions per case was seven that was derived by experiments in [13]. Zero cases reached this threshold. This generates a value of 1.

5.2 II - Similar retained queries

The chosen threshold was defined as 80 %. This was detected within prior experiments based on development of similarity measures. When using a high value such as 95 %, zero similar queries would occur. Within Fig. 1, a (WPF) page depicts the counting process of subindex II.

28 similar retained queries were achieved within 498501 query comparison iterations. This implies a temporal value of $\frac{28}{498,501}$, which will be subsequently subtracted from 1. Therefore, the value within this subindex results into $\frac{498,473}{498,501}$.

5.3 III - Missing values

In fact, the training set of the actual approach is complete concerning the values. Each tuple contains a value for each column. Zero missing values occurred within the data. This generates an excellent subindex III, value 1.

5.4 Using the main index

To avoid falling into oblivion, the training set is complete without identical cases. This refers to a high quality concerning the case base prior to an assessment of the quality.

$$\text{CBQ} = 100 \cdot \left(1 \cdot \frac{1}{3} + \frac{498473}{498501} \cdot \frac{1}{3} + 1 \cdot \frac{1}{3}\right) \tag{5}$$

In this application, the case base quality index refers to 99.9981277202.

5.5 Experiments with weights

In experiments concerning similarity measures, it was observed that only the attribute gender should be weighted with $\frac{1}{3}$. Otherwise, a simple similarity measure, which uses only a few attributes could increase or decrease the value of the result too much. Therefore, all attributes (except gender) are using the weight 1.

All subindices were associated with a weight of $\frac{1}{3}$ within the main index. In this case, increasing the weight for subindex II would decrease the index value. In another point

Fig. 1 Loaner 0.4 α - Measurement index II

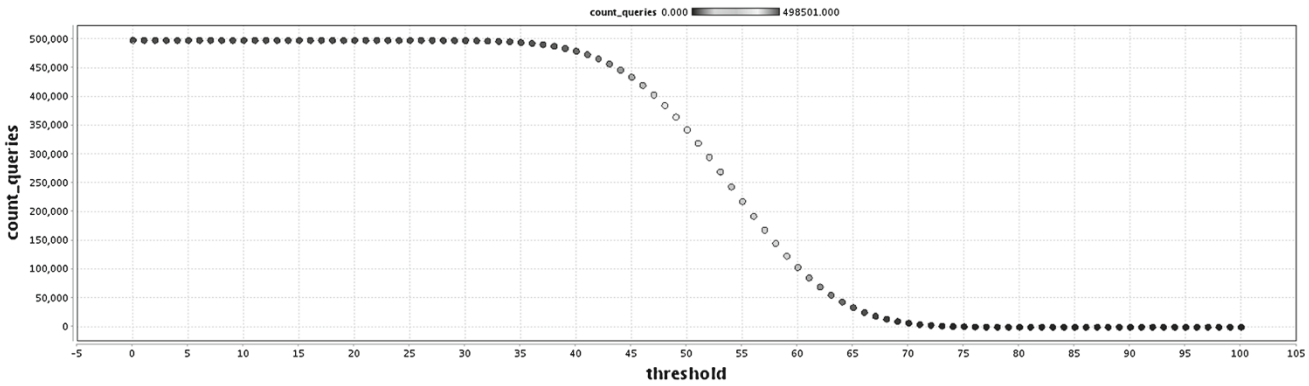


Fig. 2 Plot thresholds 0 to 100

of view when considering additional data with missing values, this would wrongly increase the index value. Therefore, a cautious weighting was applied. When using another weight for subindex II such as $\frac{5}{6}$, the value of main index is marginally modified to 99.9953193006. $\frac{5}{6}$ would be a too high value for a subindex, but in this case the result of the main index is not really affected because the associated value of the subindex was rather high $\left(1 - \frac{28}{498501}\right)$.

6 Discussion

This section provides a few notes about circumstances concerning the prototype Loaner and explanations with respect to the quality index. Concerning subindex III, the natural assumption for this index is that an application code prevents to store cases with primarily null values. Otherwise, bad case-based reasoning results would occur beside of low values in subindex III. Within an interval [0,100], thresholds were tested against the case base to see various similarity values. Within Fig. 2, thresholds and an associated count of similar query comparisons are presented. The ordinate presents the count of query comparisons from 0 to 498501. The abscissa presents thresholds from 0 to 100.

Within the threshold interval [0, 100], the plot above presents that 57 % is a point to distinguish between the nearest queries and not related queries. Concerning subindex II, 80 % was used because a threshold lower than 60 % would deliver many queries related to the concrete example within Loaner. For instance, the threshold 57 % refers to a count of 168,570 queries. To use an adequate threshold for subindex II, the concrete data such as a comma separated value file have to be analyzed. To give an excerpt within the higher threshold values regarding the second subindex, a few relations are stated as follows.

- Threshold $\rightsquigarrow$ Count queries
- 75 $\rightsquigarrow$ 710
- 76 $\rightsquigarrow$ 407
- 77 $\rightsquigarrow$ 220
- 78 $\rightsquigarrow$ 117
- 79 $\rightsquigarrow$ 65
- 80 $\rightsquigarrow$ 28
- 81 $\rightsquigarrow$ 11
- 82 $\rightsquigarrow$ 6
- 83 $\rightsquigarrow$ 3
- 84 $\rightsquigarrow$ 2
- 85 $\rightsquigarrow$ 0

In addition, it is clearly presented that a percentage of 100 refers to zero similar retained queries. Therefore, 100 % is not suitable as threshold when using a similarity measure. Another point of view, a similarity with 100 % would be identical tuples, which has to be avoided when inserting data into a schema. In Fig. 3, thresholds within the range [50, 85] are depicted, which states an excerpt of the first scatter plot. The count of similar query comparisons starts with 0 and ends with 343,038. When comparing this range to the full query range within the first scatter plot, it is clearly stated that within the range [50, 85] a higher variability occurs concerning the similar query comparisons.

The second scatter plot presents that similarity values are reduced with various different steps in a range 50–85. Within Loaner, different similarity measures are using various attributes. For gaining the similarity value concerning subindex II, a similarity measure was applied, which uses all attributes. Those are age, credit amount, credit duration, number of people liable, other installment plans, gender, personal state, purpose of the loan, credit history, employment duration, job level, other credits, duration of the current residence, installment rate concerning disposable monthly income to give an excerpt. When using all attributes, no aspect such as personal-related issues (age, gender) or credit-related considerations (credit history, credit amount) will be neglected. Subindex II calculated 28 similar retained queries within 498,501 unique comparisons between different queries. Identical tuples are not persisted. Reflexive comparisons are

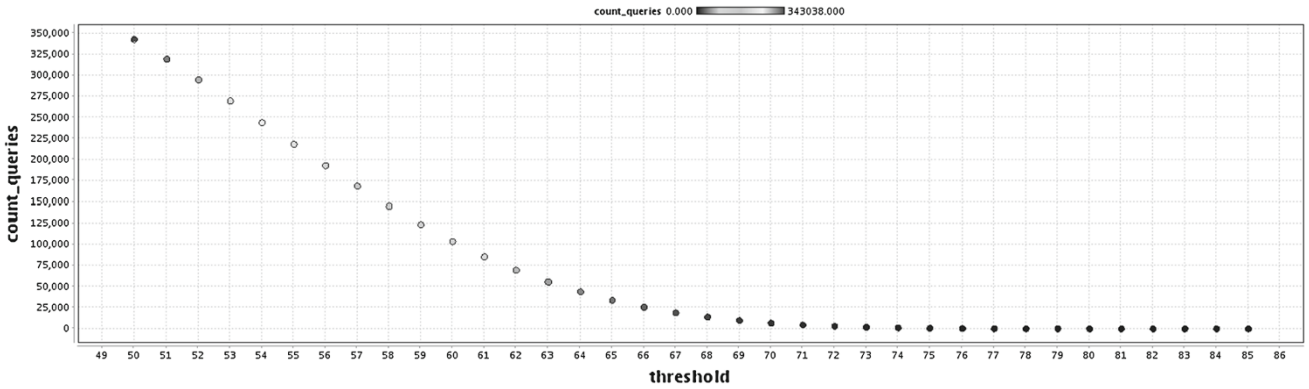
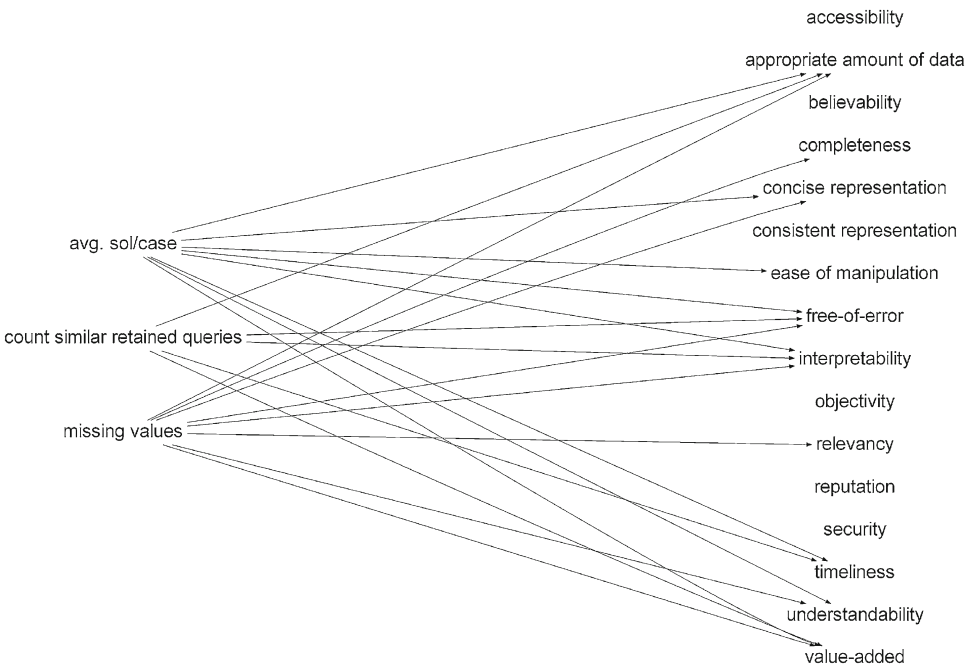


Fig. 3 Scatter plot thresholds 50–85

Fig. 4 Criteria graph



avoided. Double comparisons are avoided in addition. For instance, the similarity between query id 100 and id 770 is calculated, but not vice versa.

A second data set (called Bene) was integrated into Loaner and analyzed. The additional data set was retrieved by the author of [3]. It was more numeric-based and contained more tuples in comparison to the first one (also known as German data set [9]). The Bene data set was evaluated with the same similarity measures, which are applied towards the German data set. The Bene data set contained non-redundant solutions per case, only marginal similar retained queries and no missing values.

$$CBQ = 100 \cdot \left(1 \cdot \frac{1}{3} + \left(1 - \frac{193931}{4871881}\right) \cdot \frac{1}{3} + 1 \cdot \frac{1}{3}\right) \quad (6)$$

Hence, the result was stated as 98.6731271419 per cent. For determining the similar retained queries, 80 per cent was applied again as similarity value towards all query comparisons within the case base—except reflexive and redundant (id 31⇝94, but without 94⇝31 for instance) comparison steps.

7 Comparison to other criteria

In this section, a comparison depicts the relations between the three subindices demonstrated in this paper and criteria given by [21]. In addition, nodes/criteria without edges/relations are also plotted in Fig. 4 to provide a big picture regarding this comparison. The three subindices of this paper/CBR nodes are printed on the left side while the other criteria—according to [21]—are itemized on the right site.

To state a few relations, the completeness criterion refers to CBR counterpart ‘missing values’. But it does not lead to

the redundancy criteria regarding solutions and queries. That can be seen within a simple reason, which appears in [1]—a case cannot exist without a query and a case cannot exist without a solution. Hence, completeness refers exclusively to the third CBR subindex (missing values) because the first and second subindices are fulfilled regarding completeness from a case-based reasoner's point of view. Due to the consideration of solution retaining with respect to a case/cases, concise representation leads to the average solutions per case criterion. It does not lead to the 'count similar retained queries' criterion because this subindex focuses on the case base and not at a case level. Ease of manipulation relates to average solutions per case because this (more than one solution) can be seen as an additional requirement for CBR systems. Not every CBR system supports more than one solution per case. Free-of-error is affected by both redundancy and missing information. Missing values could cause errors when applying an impaired cleansing function for instance. Errors concerning the first subindex could occur more than once due to partial redundancy. Too many solutions per case affect the interpretability. In addition, if the count of similar retained queries would (rapidly) grow over a certain threshold value, then this could affect the interpretability concerning a case base. Obviously, missing values are impairing the relevance of data. Timeliness can be seen as correlated with the redundancy aspect, the first and second subindex, on the basis that old redundant information is more detrimental than missing values with respect to time. The understandability criterion relates to the missing values index and to the average solutions per case index.

8 Invocation of subindices

In this section, an excerpt of researched configurations towards the data quality index can be seen within Fig. 2. Within iterations and with different weights (0–$\frac{1}{1}$ in $\frac{1}{10}$ steps) per subindex, various results are gained, but as previously discussed only high data quality values are obtained due to the data sets, which are containing non-redundant and complete cases (Table 2).

Both the German and the Bene data set are evaluated when applying the three subindices. Noteworthy, weights are adjusted in different ways within different rows. In the first, second and third rows, the three subindices are presented with respect to both data sets. The threshold values are stated according to previous notes. Subsequently, all subindices are calculated with respect to weighting. The fractions of weights can be seen in the column named *note*. In contrast to a subindex, the main index refers to all subindices. Due to this weighting, it is possible to restrict the main index to a concrete subindex—for instance, when assigning 0, 1, 0 as weights. In another point of view, weights such as $\frac{1}{6}$, $\frac{2}{3}$, $\frac{1}{6}$ are implying $\frac{2}{3}$ towards subindex II and $\frac{1}{6}$ for subindex I and subindex III. Arbitrary weights can be chosen that depend on the required/desired points, which should be evaluated in a case base.

9 Big data quality

In 2011 and mainly in 2012, the term *big data* have started to occur on a frequent basis. Software applications have been established such as Apache Hadoop and MongoDB. A major player such as Oracle (Corporation) has been participated within their own product. In [6], they stated *value* (economic value) as an additional criterion for defining big data. Hence, we have variety, volume and velocity to decide if big data occur. In addition, value states the essential impact concerning data analysis. Due to the evolution of big data, which implies an increasing amount of data, it is not obvious that data quality should be restricted to a single database/node. Hence, the application of the presented data quality index is enhanced towards several nodes.

Table 2 Various weights for different configurations

Index	Aspect	Result German data	Result bene data	Note
I	Solutions per case	1	1	Threshold 7
II	Similar retained queries	$\frac{498473}{498501}$	$\frac{4677950}{4871881}$	Threshold 80
III	Missing values	1	1	Every field considered
Main	All	99.9981277202	98.6731271419	Weight $\frac{1}{3}$ for each aspect
Main	All	99.9962554404	97.3462542838	Weights $\frac{1}{6}$, $\frac{2}{3}$, $\frac{1}{6}$
Main	All	99.9943831607	96.0193814257	Weights 0, 1, 0 subindex II
Main	All	1	1	Weights 0, 0, 1 subindex III

When considering a count of n nodes/servers, then an additional weighting procedure can be applied. This copes with various different main indices, which are distributed on several servers. In CBQ (case base quality), the subscript *all* refers to all servers. The current iteration i refers to a concrete node. Hence, CBQ_i represents the main index for a single node. With respect to the main index, two equations are demonstrated.

$$\text{CBQ}_{all} = \sum_{i=1}^{n} \text{CBQ}_i \times \text{Weight}_i \tag{7}$$

Prior to this calculation, every node must provide two values—CBQ_i and the count of cases. Subsequently and posterior to a retrieval of all case counts, the weight for a node can be calculated.

$$Weight_i = \frac{\text{CaseCount}_i}{\sum_{i=1}^{n} \text{CaseCount}_i} \tag{8}$$

When applying this central computation, two features are given. In CBQ_{all}, a case base quality index is given for the entire database. For each node, the main index can be obtained that is provided by CBQ_i.

10 Conclusion

Within Loaner, the application regarding subindices I and III was expeditiously achieved due to complete training sets. Subindex II required an implementation, which refers to similarity measures. To avoid overlooking about similarities within queries, all attributes are applied to consider different aspects within a loan application. Concerning the theory, the three subindices are easy to use. When using weighting with the index formulae described above, agility can be attached to fit specific requirements of a given domain. In this approach, the weighting of the subindices within the formulae above was stated with $\frac{1}{3}$. For subindex II, a generic threshold cannot be inferred due to many different domains, which are suitable for case-based reasoning. Those are car mechanic, structural health monitoring, employee support, call center tools and text retrieval software for instance to refer to this diversity. To infer this approach within three steps namely The Good, the Bad and the Ugly, a few points are provided [17].

– The good

1. it clearly presents an index within a defined interval [0, 100]
2. in addition, the index can be easily applied to several nodes when considering big data quality
3. in the big picture concerning data quality, the demonstrated index reflects many aspects

– the Bad—even a generic index needs implementation effort
– the Ugly—using wrong weights to hide weakness of a case base would be theoretically possible

1. but that must be pragmatically seen within the competence of a domain expert who copes with that and subsequently avoids wrong weighting when remembering foundations about expert systems such as [15]
2. application of different weighting options as depicted in this paper avoids practically overlooking a potential weakness from a case-based reasoner's point of view

Big data can be applied to CBR, but not using an index concerning the case base quality could lead to obstacles. Especially, if a deletion strategy was not applied within a CBR approach, a case base with redundant and unused cases impairs the performance in reasoning processes. The proposed index can be applied to prevent these performance obstacles.

Noteworthy, changes are inevitable in the long term—the three *v* criteria regarding big data are enhanced with a fourth *v* criterion—*value*.

In the end, the comparison of data quality criteria provided an insight. We can distinguish into (at least) three classes of criteria—to name it briefly, relation, data generation and implementation. Relation—a criterion that is coupled with another aspect/other aspects such as missing values $\rightsquigarrow$ appropriate amount of data. Data generation—in contrast, a criterion, such as reputation, that is coupled with the sourcing process/data generation and decoupled from other data quality aspects. Implementation—an additional class can be seen for criteria that are decoupled from the sourcing process/data generation and not related to other aspects. Accessibility and security depict examples for the third class.

11 Future work

To gain results/values as input towards the data quality index, an essential feature was the application of similarity measures within Loaner. Those have coped with numerical and nominal attributes in both data sets. To be more generic with respect to future case-based reasoning systems, an interchange format would be required to define these measures. This similarity measure definition language has to include all required information when providing a similarity measure for a case-based reasoning system. The definition of this kind of language would improve interchange between case-based reasoning

systems and an increased velocity during development of a measure and reducing both labor effort/cost.

Acknowledgments Appreciation goes to the reviewers for their comprehensive hints and feedback. Thanks to Prof. Baesens for providing an additional data set to enhance the research for both thesis and this paper.

Open Access This article is distributed under the terms of the Creative Commons Attribution License which permits any use, distribution, and reproduction in any medium, provided the original author(s) and the source are credited.

References

1. Aamodt, A., Plaza, E.: Case-based reasoning: foundational issues, methodological variations, and system approaches. AI Commun. **7**(1), 39–59 (1994)
2. Arshadi, N., Jurisica, I.: Maintaining case-based reasoning systems: a machine learning approach. In: ECCBR. pp. 17–31 (2004)
3. Baesens, B., Setiono, R., Mues, C., Vanthienen, J.: Using neural network rule extraction and decision tables for credit-risk evaluation. Manag. Sci. **49**(3), 312–329 (2003)
4. Basili, V.R.: The Experience factory and its relationship to other improvement paradigms. In: Sommerville, I., Paul, M. (eds.) Software Engineering-ESEC, '93, 4th European Software Engineering Conference. Springer, Berlin, Heidelberg, pp 68–83
5. Bergmann, R., Althoff, K.D., Breen, S., Göker, M., Manago, M., Wess, S.: Developing industrial case based reasoning applications: the INRECA methodology, vol. Lecture Notes in Artificial intelligence Berlin, LNAI 1612, Berlin. Springer Verlag (2003)
6. Corporation, O.: Big data for the enterprise (2013), report
7. DeCoste, D.: The future of chess-playing technologies and the significance of kasparov versus deep blue. In: Papers from the 1997 AAAI Workshop (1997)
8. Ferrucci, D.A.: Ibm's watson/deepqa. SIGARCH computer architecture news 39(3) (2011)
9. Frank, A., Asuncion, A.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2010)
10. Hönigl, J.: Case-based reasoning with respect to banking driven by knowledge discovery. Ph.D. thesis, Johannes Kepler University (2014)
11. Hönigl, J., Kosorus, H., Küng, J.: On reasoning within different domains in the past, present and future. In: 23rd Database and expert systems applications (DEXA), 2012. 2nd International Workshop on Information systems for situation awareness and situation management-ISSASiM'12 (2012)
12. Hönigl, J., Küng, J.: A data quality index with respect to case bases within case-based reasoning. In: Proceedings of ACIIDS 2014—The 6th Asian Conference on Intelligent information and database systems. Springer (2014)
13. Hönigl, J., Nebylovych, Y.: Building a financial case-based reasoning prototype from scratch with respect to credit lending and association models driven by knowledge discovery. Central & Eastern European Software Engineering Conference in Russia (2012)
14. Jurisica, I., Nixon, B.: Building quality into case-based reasoning systems. In: Pernici, B., Thanos, C. (eds.) Advanced Information Systems Engineering. Lecture Notes in Computer Science, vol. 1413, pp. 363–380. Springer, Berlin (1998)
15. Klein, M.R., Methlie, L.B.: Knowledge-based decision support systems with applications in business. Wiley, England (1995)
16. Leake, D.B.: Cbr in context: The present and future. In: Reasoning from remindings. pp. 3–30. MIT Press, Menlo Park, California (1996)
17. Leone, S.: The good, the bad and the ugly. il buono, il brutto, il cattivo. (original title) (1966)
18. Miller, G.A.: The magical number seven, plus or minus two some limits on our capacity for processing information. Psychol. Rev. **63**, 81–97 (1956)
19. Newborn, M. (ed.): Deep blue establishes historic landmark. In: Beyond Deep Blue, pp. 1–26. Springer, London (2011)
20. Pan, R., Yang, Q., Pan, S.J.: Mining competent case bases for case-based reasoning. Artif. Intel. **171**(16–17), 1039–1068 (2007)
21. Pipino, L.L., Lee, Y.W., Wang, R.Y.: Data quality assessment. Commun. ACM **45**(4), 211–218 (2002)
22. Rahul Premraj, M.S.: Assessing case base quality. Bournemouth University and Brunel University (2005)
23. Reinartz, T., Iglezakis, I.: On quality measures for case base maintenance. In: Proceedings of the 5th European Workshop on case-based reasoning. pp. 247–259. Springer-Verlag (2000)
24. Roth-Berghofer, T., Reinartz, T.: Mama: a maintenance manual for case-based reasoning systems. In: ICCBR. pp. 452–466 (2001)
25. Tsatsoulis, C., Cheng, Q., Wei, H.Y.: Integrating case-based reasoning and decision theory. IEEE Expert **12**(4), 46–55 (1997)